

## **REMARKS/ARGUMENTS**

Claims 1-6, 23, 25-27, and 37-46 are pending in the present application. In this amendment, claims 1-6, 23, 25-27, 39, 42, and 43 are canceled; claims 37, 38, 40-41, 44, 45, and 46 are amended; and claims 47 and 48 are added. Reconsideration of the claims is respectfully requested.

Applicants are not conceding in this application that the claims as presented prior to this amendment are not patentable. The present claim amendments are only for facilitating expeditious prosecution of the claims. Applicants respectfully reserve the right to pursue these and other claims in one or more continuations and/or divisional patent applications. Support for the amendments to the claims may be found in the claims as originally presented, as well as in the following portions of the specification: [0037]; [0038]; [0039]; [0040]; [0041]; [0042]; [0043]; [0044]; [0045]; [0046]; [0047]; [0048]; [0049]; [0050]; [0051]; [0052]; Figure 6; and Figure 7.

### **I. 35 U.S.C. § 102, Anticipation**

The Examiner has rejected claims 37-41 and 44-46 under 35 U.S.C. § 102 as being anticipated by Hitz, U.S. Patent Application Publication No. 2002/0091670 (hereinafter “Hitz”). This rejection is respectfully traversed.

A prior art reference anticipates the claimed invention under 35 U.S.C. § 102 only if every element of a claimed invention is identically shown in that single reference, arranged as they are in the claims. *In re Bond*, 910 F.2d 831, 832, 15 U.S.P.Q.2d 1566, 1567 (Fed. Cir. 1990). All limitations of the claimed invention must be considered when determining patentability. *In re Lowry*, 32 F.3d 1579, 1582, 32 U.S.P.Q.2d 1031, 1034 (Fed. Cir. 1994). Anticipation focuses on whether a claim reads on the product or process a prior art reference discloses, not on what the reference broadly teaches. *Kalman v. Kimberly-Clark Corp.*, 713 F.2d 760, 218 U.S.P.Q. 781 (Fed. Cir. 1983). In this case each and every feature of the presently claimed invention is not identically shown in Hitz, arranged as they are in the claims, and, accordingly, Hitz does not anticipate the claims.

**A. Claim 37**

In rejecting claim 37, the Examiner states:

(Claims 37, 44)

Hitz discloses a computer implemented method comprising receiving a command to store a file in a file system having an inode ("on-disk inode 310", paragraph [0053]), wherein the inode is usable to store metadata associated with the file ("The inode information 310A comprises information about the owner of a file, permissions, file size, access time, etc. that are well-known to a person skilled in the art", paragraph [0053]); responsive to the file having a size that is greater than a block size of blocks in the file system ("For a file having a size that is greater than or equal to 64 KB and is less than 64 MB", paragraph [0056]), storing data of the file only in an integer number of blocks ("The block number entries 0-15", paragraph [0055]), wherein a remainder data of the file remains after storing, and wherein the remainder data is less than the block size; and placing the remainder data directly into the inode ("For a small file having a size of 64 bytes or less, data is stored directly in the inode itself instead of the 16 block numbers", paragraph [0054]).

Office Action dated April 16, 2009, pp. 2-3.

Claim 37, as amended herein, is as follows:

37. (Currently Amended) A computer implemented method comprising:  
receiving, using a processor, a command to store a file in a file system having an inode, wherein the inode is usable to store metadata associated with the file;  
responsive to the file having a size that is greater than a block size of blocks in the file system, storing a plurality of data of the file only in an integer number of blocks, wherein a first remainder data of the file remains after storing, and wherein the first remainder data is less than the block size;  
placing the first remainder data directly into the inode; and  
responsive to a second remainder data of the file still remaining after the first remainder data is placed into the inode, placing the second remainder data into an unused space of a partially used block of the file system, wherein the partially used block also stores data of another file.

Contrary to the Examiner's assertions, Hitz does not anticipate amended claim 37. Specifically, Hitz fails to disclose or suggest the amended claim 37 feature of "responsive to second remainder data of the file still remaining after the first remainder data is placed into the inode, placing the second remainder data into an unused space of a partially used block of the file system, wherein the partially used block also stores data of another file."

The amended claim 37 feature of "responsive to second remainder data of the file still remaining after the first remainder data is placed into the inode, placing the second remainder data into an unused space of a partially used block of the file system, wherein the partially used

block also stores data of another file,” is drawn from the subject matter recited by cancelled claim 39. In the Office Action dated April 16, 2009, the Examiner stated the following with regard to claim 39:

(Claims 39, 45)

Hitz discloses responsive to second remainder data of the file still remaining after the remainder data is placed into the inode, placing the second remainder data into an unused space of a first block of the file system, wherein the first block also stores data of another file, and wherein the first block comprises a last block of the another file (“For a file having a size that is greater than or equal to 64 KB and is less than 64 MB, each of the 16 block numbers references a single-indirect block. In turn, each 4 KB single-indirect block comprises 1024 block numbers that reference 4 KB data blocks. FIG. 4C is a diagram illustrating a Level 2 inode 310 comprising 16 block numbers 310B that reference 16 single-indirect blocks 430A-430C. As shown in FIG. 4C, block number entry 0 points to single-indirect block 430AV, paragraph [0056]).

Office Action dated April 16, 2009, p. 3.

The Examiner believes that the claim 39 feature, from which the amended claim 37 feature of “responsive to second remainder data of the file still remaining after the first remainder data is placed into the inode, placing the second remainder data into an unused space of a partially used block of the file system, wherein the partially used block also stores data of another file,” is drawn, is disclosed in the following portion of Hitz:

[0056] For a file having a size that is greater than or equal to 64 KB and is less than 64 MB, each of the 16 block numbers references a single-indirect block. In turn, each 4 KB single-indirect block comprises 1024 block numbers that reference 4 KB data blocks. FIG. 4C is a diagram illustrating a Level 2 inode 310 comprising 16 block numbers 310B that reference 16 single-indirect blocks 430A-430C. As shown in FIG. 4C, block number entry 0 points to single-indirect block 430A. Single-indirect block 430A comprises 1024 block numbers that reference 4 KB data blocks 440A-440C. Similarly, single-indirect blocks 430B-430C can each address up to 1024 data blocks.

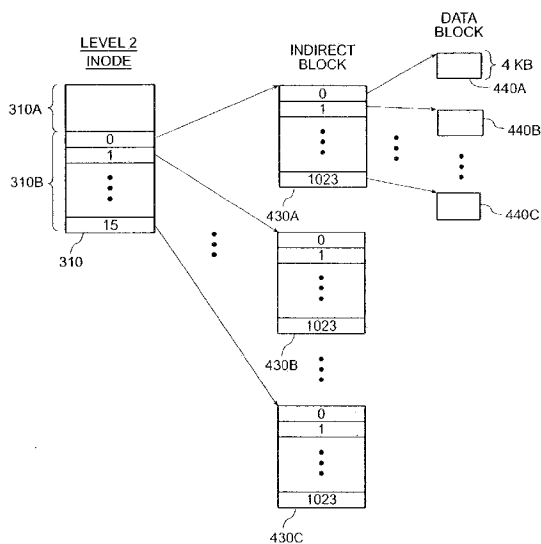
Hitz, [0056].

Hitz is directed generally to keeping files in a file system in a consistent state and creating read-only copies of files. Hitz discloses a “write anywhere file system” that contains a root inode at a fixed location on disk. The root inode contains some metadata about data in the file system and points to an inode file, elsewhere on disk. The inode file contains a sequence of all inodes on the disk. The inodes stored in the inode file are used to point to the various blocks on disk where a particular file or directory may be found. Hitz discloses that inodes are of a static size on disk, for

example, 4KB. Since the inode is 4KB large, it can contain the addresses of 16 blocks on disk where data may be found for the file or directory represented by the inode.

Hitz also discloses that inodes can have one or more levels of indirection. According to Hitz, a 4KB inode can only hold the addresses for 16 blocks on disk, that is, the data for the file represented by the inode. Hitz discloses that this limitation can be increased by making each of the 16 slots in the inode point to an indirect block, instead of the actual data for the file. Assuming a file of the maximum size addressable by Hitz, each of the 16 slots in an inode will point to an indirect block. Each indirect block contains the addresses of 16 blocks on disk that make up the data for the file represented by the inode. By using indirect files, Hitz discloses that the maximum file size can be increased.

The portion of Hitz reproduced above and cited by the Examiner as disclosing the amended claim 37 feature of “responsive to second remainder data of the file still remaining after the first remainder data is placed into the inode, placing the second remainder data into an unused space of a partially used block of the file system, wherein the partially used block also stores data of another file,” discloses a specific example of using an indirect block to increase the maximum file size of the file system. The above-reproduced portion of Hitz discloses that if a file is of a size less than 64MB, but greater than or equal to 64KB, a single indirect block can be used to store the file in the “write anywhere file system.” Figure 4C of Hitz illustrates the portion cited by the Examiner:



Hitz, Fig. 4C.

In the figure reproduced above, Hitz shows how an indirect block can be used to increase the maximum file size in the file system. The Level 2 Inode 310 does not store the precise memory addresses of the data for the file represented by Level 2 Inode 310. Instead, Level 2 Inode 310 stores addresses 310B for indirect blocks 430A, 430B, and 430C. Each indirect block stores some of the memory addresses of the blocks that contain the actual data for the file represented by Level 2 Inode 310. In this case, indirect block 430A contains the memory addresses for data blocks 440A, 440B, and 440C.

Contrary to the Examiner's assertion, Hitz does not disclose or suggest "responsive to second remainder data of the file still remaining after the first remainder data is placed into the inode, placing the second remainder data into an unused space of a partially used block of the file system, wherein the partially used block also stores data of another file," at least because Hitz does not store the remainder data into an unused space of a partially used block of the file system. As discussed above, the portion of Hitz cited by the Examiner as disclosing this feature of amended claim 37 only discloses that indirect blocks can be used to help address the data blocks of files larger than those that can be addressed using a single inode. In amended claim 37, second remainder data that remains after the remainder data is placed into the inode is placed into an *unused space of a partially used block* of the file system. The partially used block also stores data of another file. Therefore, Hitz does not disclose or suggest "responsive to second remainder data of the file still remaining after the first remainder data is placed into the inode, placing the second remainder data into an unused space of a partially used block of the file system, wherein the partially used block also stores data of another file."

Since claims 38, 40 and 41 depend from amended claim 37, claims 38, 40, and 41 recite at least the same distinctions over Hitz as amended claim 37. Claims 38, 40, and 41 also recite other additional combinations of features not suggested by the reference. Additionally, amended claim 44 and claim 47 recite similar features as claim 37. Applicants submit that amended claim 44 and claim 47, as well as dependent claims 44 and 45, are allowable for at least the same reasons as amended claim 37. Consequently, it is respectfully urged that the rejection of claims 37-41 and 44-46 have been overcome.

## B. Claim 38

In rejecting claim 38, the Examiner states:

(Claim 38)

Hitz discloses wherein placing further comprises placing the remainder data in an extension area of the inode, wherein the extension area was formerly reserved for the metadata ("The inode information 310A comprises information about the owner of a file, permissions, file size, access time, etc. that are well-known to a person skilled in the art", paragraph [0053]).

Office Action dated April 16, 2009, p. 3.

Amended claim 38 recites the following:

38. The computer implemented method of claim 37 wherein placing the first remainder data directly into the inode further comprises placing the first remainder data in an extension area of the inode formerly reserved for the metadata, and the computer implemented method further comprises:[.]

updating a mode field in the inode to designate that the first remainder data of the file has been stored in the inode;

prior to performing the step of placing the second remainder data into an unused space of the partially used block of the file system, determining whether the partially used block exists and whether the partially used block has a sufficient free space to store the second remainder data;

updating the mode field in the inode to designate that the second remainder data has been stored in the partially used block; and

responsive to the partially used block becoming full from storing the second remainder data in the partially used block, removing the partially used block from a list of free shared blocks, wherein, for each free shared block on the list of free shared blocks, the list of free shared blocks contains a block number, a free byte quantity, and a pointer to a next free shared block.

Contrary to the Examiner's assertion, Hitz fails to anticipate claim 38 at least because Hitz does not disclose or suggest at least the amended claim 38 feature of "wherein placing the first remainder data directly into the inode further comprises placing the first remainder data in an extension area of the inode formerly reserved for the metadata." Additionally, Applicants point out that Hitz fails to disclose or suggest the features of amended claim 38 added in this amendment.

**1. Hitz fails to disclose or suggest "wherein placing the first remainder data directly into the inode further comprises placing the first remainder data in an extension area of the inode formerly reserved for the metadata."**

Hitz fails to anticipate amended claim 38 at least because Hitz fails to disclose or suggest "wherein placing the first remainder data directly into the inode further comprises placing the

first remainder data in an extension area of the inode formerly reserved for the metadata.” The Examiner disagrees, citing to the following portion of Hitz:

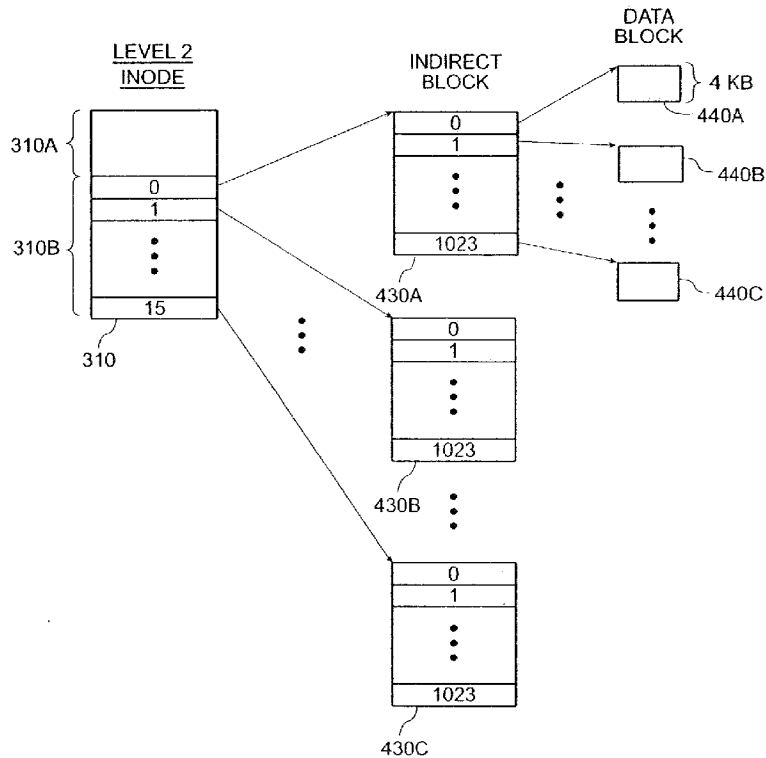
[0053] WAFL inodes are distinct from prior art inodes. Each on-disk WAFL inode points to 16 blocks having the same level of indirection. A block number is 4-bytes long. Use of block numbers having the same level of indirection in an inode better facilitates recursive processing of a file. FIG. 3 is a block diagram illustrating an on-disk inode 310. The on-disk inode 310 is comprised of standard inode information 310A and 16 block number entries 310B having the same level of indirection. *The inode information 310A comprises information about the owner of a file, permissions, file size, access time, etc. that are well-known to a person skilled in the art.* On-disk inode 310 is unlike prior art inodes that comprise a plurality of block numbers having different levels of indirection. Keeping all block number entries 310B in an inode 310 at the same level of indirection simplifies file system implementation.

Hitz, [0053] (emphasis added).

As discussed in Section I.A., *supra*, Hitz is directed generally to keeping files in a file system in a consistent state and creating read-only copies of files. The above-reproduced portion of Hitz discloses that WAFL (“write anywhere file system”) inodes in a file system must have a common level of indirection. For example, if one WAFL inode in Hitz has one level of indirect blocks, then all WAFL inodes must have one level of indirect blocks for efficiency purposes.

The portion also discloses that the inode comprises standard inode information in addition to the block number entries, that is, the data block or indirect block addresses. Hitz discloses that the standard inode information is information about the owner of a file, permissions, file size, and access time. This is not the same as the amended claim 38 feature of “wherein placing the first remainder data directly into the inode further comprises placing the first remainder data in an extension area of the inode formerly reserved for the metadata,” at least because Hitz only discloses that metadata exists in an inode, and does not place remainder data in an extension area of the inode as recited by amended claim 38.

In fact, Hitz explicitly teaches away from amended claim 38. Examine the following figure:



Hitz, Fig. 4C.

In the above-reproduced figure, Hitz discloses that data is stored in data blocks, such as 440A, 440B, and 440C. Reference numeral 310A refers to the standard inode information, or metadata, described above. Hitz does not store file data in the extension area of the inode that was formerly reserved for metadata. Therefore, Hitz cannot disclose or suggest the amended claim 38 feature of “wherein placing the first remainder data directly into the inode further comprises placing the first remainder data in an extension area of the inode formerly reserved for the metadata.”

Amended claim 45 recites similar features as amended claim 38, and Applicants believe that amended claim 45 is patentable over the cited art for at least the same reasons as claim 38. Therefore, Applicants request the Examiner withdraw the rejection of claims 38 and 45 under 35 U.S.C. § 102.



## **2. Hitz fails to disclose or suggest the additional features recited in amended claim 38**

Hitz also fails to anticipate amended claim 38 because Hitz fails to disclose or suggest the additional features of amended claim 38 added in this amendment. Recall that claim 38 recites the following:

38. The computer implemented method of claim 37 wherein placing the first remainder data directly into the inode further comprises placing the first remainder data in an extension area of the inode formerly reserved for the metadata, and the computer implemented method further comprises: [[.]]  
updating a mode field in the inode to designate that the first remainder data of the file has been stored in the inode;  
prior to performing the step of placing the second remainder data into an unused space of the partially used block of the file system, determining whether the partially used block exists and whether the partially used block has a sufficient free space to store the second remainder data;  
updating the mode field in the inode to designate that the second remainder data has been stored in the partially used block; and  
responsive to the partially used block becoming full from storing the second remainder data in the partially used block, removing the partially used block from a list of free shared blocks, wherein, for each free shared block on the list of free shared blocks, the list of free shared blocks contains a block number, a free byte quantity, and a pointer to a next free shared block.

Hitz does not disclose or suggest “updating a mode field in the inode to designate that the first remainder data of the file has been stored in the inode,” “prior to performing the step of placing the second remainder data into an unused space of the partially used block of the file system, determining whether the partially used block exists and whether the partially used block has a sufficient free space to store the second remainder data,” “updating the mode field in the inode to designate that second remainder data has been stored in the partially used block,” or “responsive to the partially used block becoming full from storing the second remainder data in the partially used block, removing the partially used block from a list of free shared blocks, wherein, for each free shared block on the list of free shared blocks, the list of free shared blocks contains a block number, a free byte quantity, and a pointer to a next free shared block.”

Generally, the features recited above are directed to updating the mode field of the inode, making determinations regarding the partially used block, and removing the partially used block from the list of free blocks. Hitz fails to disclose or suggest any of these features of amended claim 38. For example, Hitz is silent to updating the mode field in the inode. Also, since Hitz teaches away from the partially used block as recited by amended claim 38, Hitz cannot disclose

making a determination regarding the partially used block. Finally, Hitz does not disclose the list of free shared blocks, at least because Hitz does not disclose free shared blocks as recited in amended claim 38.

Amended claim 45 and claim 48 recite similar features as amended claim 38, and Applicants believe that amended claim 45 and claim 48 are patentable over the cited art for at least the same reasons as claim 38. Therefore, Applicants request the Examiner withdraw the rejection of claims 38 and 45 under 35 U.S.C. § 102.

## **II. Conclusion**

It is respectfully urged that the subject application is patentable over Hitz and is now in condition for allowance.

The Examiner is invited to call the undersigned at the below-listed telephone number if in the opinion of the Examiner such a telephone conference would expedite or aid the prosecution and examination of this application.

DATE: July 16, 2009

ROS/ek

Respectfully submitted,

/Rudolf O. Siegesmund/

Rudolf O. Siegesmund  
Reg. No. 37,720  
Yee & Associates, P.C.  
P.O. Box 802333  
Dallas, TX 75380  
(972) 385-8777  
Attorney for Applicants